THE ALOG DISPLAYMAKER

from

ALOG COMPUTING

1040 Veronica Springs Road
Santa Barbara, CA 93105

Program and Manual Contents (C) 1983 D. Higgins

Created in whole or part using valFORTH products of
Valpar International, Tucson, AZ 85713 USA
Based on fig-FORTH, provided through the courtesy of
Forth Interest Group, P.O. Box 1105, San Carlos, CA 94070

[This entire manual was prepared using the ALOG PAGEWRITER
and the ALOG DISPLAYMAKER]

! ! ! ! WARNING ! ! ! !

COPYING THE ALOG DISPLAYMAKER DISK

The ALOG DISPLAYMAKER program disk is not copy-protected and you may make copies (for your own use only, please !). To avoid the inconvenience of swapping disks, display screens can be stored on the program disk. This creates the possibility of some error that might wipe out the program itself. WE THUS STRONGLY SUGGEST THAT YOU IMMEDIATELY COPY THE PROGRAM DISK AND USE ONLY THE WORKING COPY FOR STORING DISPLAY SCREENS. Note that the ATARI DOS disk copy routines will copy your original disk, but you may have trouble making copies of copies !

THE ALOG DISPLAYMAKER

The ALOG DISPLAYMAKER is a utility program designed to help the user create high resolution displays on a monitor or television set being driven by an ATARI computer. The program enables one to both generate geometric figures (e.g. lines, circles, rectangles,etc.) and to easily type various characters at arbitrary locations on the screen. Characters can include the standard alphabet, in lower or upper case (including inverse letters and three letter sizes), as well as any special characters the user may wish to create (using the built-in character editor). Four colors can be displayed at one time, and each of these colors can be chosen using a visual display of all 128 colors available on the Atari. Display resolution is 160 x 160. All displays can be saved on disk, or printed in two sizes on an (optional) Epson MX-80 printer with graphics options. Displays can also be shown sequentially in a " slide show " format.

This program has many potential uses. Both children and adults may find it just plain fun to create colorful "works-of-art". The ALOG DISPLAY-MAKER can also be used for preparing computer "slides" for educational or business purposes. These slides can be re-created from disk storage or duplicated photographically with an ordinary 35 mm camera. Full or quarter page print-outs can also be made with an Epson MX-80 printer and used in reports or made into overhead transparencies.

SUMMARY OF FEATURES

Graphics Mode:

- "Rubberband" line drawing
- Joystick controlled drawing
- Adjustable ellipse drawing (includes circles)
- Rectangle drawing
- Semi-automatic Fill
- Easy color manipulation
- "Zoom" magnification for details
- "Slide Show" display presentations

Character mode (Text Input)

- Three different character sizes
- Three different colors
- Ability to mix colors, sizes, and place characters anywhere
- Includes upper case, lower case, and inverse video
- Extra character set with Greek letters
- * "Built-in" character editor *
  (Add any character you wish without interrupting program)

General

- Program disk copyable
- Store displays on program disk
- Copy display on graphics printer
  (includes a double size option that fills standard 8 1/2 x 11 page)

(ATARI is a trademark of ATARI, Inc.)

REQUIRED EQUIPMENT

ATARI 400 or 800 Computer
48K of RAM
One disk drive
Joystick (optional)
Printer with dot graphics capability (e.g. Epson with Graftrax, Prowriter,etc.)
(used to print copies of displays)

LOADING THE ALOG DISPLAYMAKER

1. Remove all cartridges from your computer and turn computer off.

2. Turn on your disk drive.

3. When the disk drive is ready, insert the ALOG DISPLAYMAKER program disk.

4. Turn on your computer and TV set or monitor.

5. The program will begin to load automatically. When loading is completed, you will first see a title screen. This will disappear in a few seconds and you will then see a cross-hair type cursor centered on the screen and a text window at the bottom containing a command summary.

AN INTRODUCTORY SLIDE SHOW

One of the special features of the ALOG DISPLAYMAKER is the ability to present a "slide show" of previously created displays. The original program disk contains eight such "slides" that can be used to show the various features of the ALOG DISPLAYMAKER. To eXhibit these displays, press the <X> key. You will first be given some preliminary instructions. You can then view the introductory screens by simply pressing the <1> - <8> keys. Press the <RETURN> key when finished to return to the main program.

Any displays you create can be shown in a similar manner once they have been saved on disk. In fact, you can prepare an art show, business presentation, or briefing using one or more disks and this "slide show " feature.

A BRIEF JOURNEY THROUGH THE ALOG DISPLAYMAKER

The ALOG DISPLAYMAKER is made up of two seperate sub-programs which will be referred to as the GRAPHICS MODE and the TEXT (or character) MODE. The program is in the graphics mode immediately after being loaded. This mode can be identified by the commands shown in the text window. The first line of this window shows:

<L>ines <D>raw <E>llipse <F>ill

The currently active command is shown in inverse video. (<L>ines should be in inverse video when the program is first loaded.) The single key indicated by <key> is used to select other commands. (Try hitting the <D>, <E>, or <F> keys.). The present color (indicated by <0>, <1>, <2>, or <3>) is also shown in inverse video.

Note that most commands in the graphics mode are controlled using the joystick (plugged into port #1). The keyboard may be used instead of the joystick by using the arrow keys to move the cursor and the pressing space bar instead of the fire button. Note that keyboard response is slower than that of the joystick, making it ideal for cases where fine positioning is needed.

<L>ines

While in the <L>ines mode, try moving the screen cursor around with the joystick (or the keyboard arrow keys). Pick a location where you want a line to start and press the fire button on the joystick (or the space bar on the keyboard). Now move the cursor away from the starting spot. You should see one cursor remain in the spot marked and another moving with the joystick. A flickering line will be shown between the two cursors. When the moving cursor is where you want the line to end, simply hit the fire button once more to "fix" the end point of the line. A permanent line will then appear on the screen. This type of line drawing is known as "rubber-band" graphics because the line is stretched like a rubber-band from the starting point to the moving cursor until put in-place with the fire button. You can then move the cursor to any location, hit the fire button to fix the starting location, and begin the entire process once again. You can change the color of the line being drawn at any time by hitting the color number keys, <0>-<3>. (Note that <0> is the background color. Lines drawn with color <0> may thus be invisible.)

<D>raw and <Z>oom

In the <D>raw mode the cursor acts like a paint brush which only "draws" on the screen when the fire button is being pushed. Note that a good way of erasing an unwanted line or some mistake is to "draw" over it with color <0>. Because zero is the background color, this will erase whatever is below the cursor. A related command, not explicitly shown in the text window, is <Z>oom. This command creates a greatly magnified picture of the region surrounding the cursor location at the time the <Z> key was pressed. One can draw on this magnified screen by positioning the cursor and pressing the fire button (or the space bar). The <Z>oom command is useful for drawing fine details. To leave the <Z>oom mode, press the key for any other legal command.

<R>ectangle and <F>ill

Other modes operate in a similar fashion. For example, in the <R>ectangle mode, one "marks" the two opposite corners of a rectangle with the cursors and the program will draw a box through the corners. The <F>ill command operates by filling in the screen along a horizontal line both to the right and to the left of the current cursor position until a color other than the current background color is reached. By holding down the fire button while moving the cursor, one can rapidly color-in large areas of the screen. (Draw a rectangle and try to color the inside.)

<E>llipses

The <E>llipse command is used to draw elliptical shapes, of which the circle is a special case. To draw a circle, first "mark" the center with one cursor and then move the other cursor out to a distance equal to the desired radius. Mark this location by pressing the fire button once again. A circle will then be drawn at the indicated location.

To change the shape of the ellipse, press the <OPTION> button while in the <E>llipse command mode. A large green rectangle will appear with crossed vertical and horizontal orange lines. The relative lengths of these two lines represent the "aspect ratio" of any ellipse which will be drawn. When they are the same length, the result is a circle. You can use the joystick to shorten or lengthen the vertical line. Imagine an ellipse centered where the lines cross and just touching the ends of the lines. You can thus judge the shape for each vertical line height. Pick one and then press the fire button (or space bar) and you will be returned to the normal graphics mode. Now try drawing an ellipse again. The shape will be circular only if the vertical and horizontal lines were the same length. (Note that when the ellipses are highly elongated, the distance between the center cursor and the outer cursor is preserved only along the horizontal direction.)

Changing <C>olors

Now consider the <C>olors command. When you press the <C> key two vertical bars will appear. The bar on the left will display 16 different colors, all at the same luminance, while the bar on the right shows the 8 luminances that can be generated for each of the 16 colors. The color value for the 8 luminances is indicated by an arrow pointing to the left bar, while the luminance of the left bar is indicated by the arrow pointing to one part of the right bar. (This description may sound a bit confusing to new Atari owners, but the visual effects are straightforward.) The graphics mode being used can display 3 foreground colors against 1 background color. Each of these 4 colors can be set to one of the 128 combinations that are possible with 16 colors, each having 8 luminance possibilities. You are first asked to choose which of the 4 screen colors you wish to change. (Remember that # 0 is the background color.) Pick 0 as an example. Then use the <OPTION> and <SELECT> keys to move the arrows down the two color bars. The background of the screen will change color as these keys are pressed. When you obtain a color you like, simply press the <START> key. You can then change one of the other color values, or quit by pressing <Q>. The best way to understand this command is simply to play around with it and see all the color combinations possible with your Atari. (Note that you can make various parts of the display "disappear" by choosing certain colors that make text or drawings the same color as the background. If this causes confusion, simply hit the <SYSTEM RESET> key.)

Display Modes

Now press the <ESC> key. The screen and text window will change colors, but other than that nothing too spectacular should happen. Press <ESC> again and the screen will return to its initial configuration. The <ESC> key toggles the internal graphics display mode being used by the Atari but uses the same display data. The original mode is often called Graphics 7+. It is a 4 color mode not available with Atari Basic. It has twice the vertical resolution of Basic's graphics mode 7 (thus 7+). Hitting the <ESC> key once changes the display mode to Basic's graphics mode 8. This display mode is described as a high resolution 1 color mode. A "trick" called "artifacting" makes it possible to produce several colors in this mode, however. (See appendix with technical information.) Being able to flip between these two types of display modes is primarily useful when small text is desired. This is further discussed later.

<S>ave, <G>et, and <P>rint

The other commands are fairly obvious. <S>ave and <G>et enable you to store and retrieve displays using your disk drive. Room for 8 screens has been alloted on the program disk. Each display is simply assigned a number as identification. The <P>rint command enables you to print a copy of the display if you have an Epson printer with graphics capability. The printed output is available in two sizes.

One command not listed in the text window is <CTRL><CLEAR> or <SHIFT><CLEAR>. As you may have guessed, this clears the graphics screen.

<T>ext or Character Input

Pressing the <T> key will shift you to the Text input mode. This mode is easily identified by the first line of the text window, which says:

TEXT INPUT Press <CTRL><Q> to quit

Try pressing the <CTRL><Q> key. This should return you to the graphics mode previously discussed. Return to the text mode again by hitting the <T> key. You will notice the outline of a rectangle in the upper left hand corner of the screen. This is a cursor which indicates where the next character typed will be placed on the screen. Try typing a few letters. You may have noticed that the size of the letters is the same as that obtained in Basic Graphics mode 1. If you wish to change the text size or color you must first press the <TAB> key and then release it. The next key pressed is then recognized as a command rather than displayed on the screen. Press <TAB> followed by <1>,<2>, or <3> to get characters in colors 1, 2, or 3. Similarly, follow the <TAB> key with <S>, <M>, or <L> to get small, medium, or large characters. The cursor will change size when you give the <S>, <M>, or <L> commands.

Moving the Text Cursor

Cursor position can be controlled using various keys, such as <RETURN>, <BACK S>, and the 4 arrow keys (i.e. <CTRL><->, <CTRL><=>, <CTRL><+>, and <CTRL><*>). Most keys operate in a manner similar to their normal operation. You will notice, however, that the up and down arrows only move the cursor a small distance up or down. This distance is smaller than the height of a character. You thus have the freedom to type a character at almost any vertical location on the screen. (This is very useful for subscripts or superscripts.) If you wish to move the cursor up or down the screen more rapidly, try <CTRL><U> or <CTRL><D>.

The best way to learn about this mode of program operation is simply to play around with it for awhile. Note that you can also type lower case and inverse video characters as well as many of the special Atari characters. Be sure to shift back to the normal upper case letters before giving any commands.

Optional Character Set

By now you may have noticed the somewhat strange shape on the right hand side of the lower text window. This shape is the Greek letter alpha; it is the first character of a special character set which can be modified to include whatever characters may be of interest. Press the <CTRL><F> key to cycle forward through this character set. (Use <CTRL><B> to go back to a character you may have passed.) If you find one you wish to display on the screen, simply stop when that character is visible and move the cursor to the location where you want the character displayed. Then press the <P> key while holding down the <OPTION> button (i.e. <OPTION><P>). This will print the optional character.

Special Character Editor

However, you may not find the special character you need, or you just may want to create some new characters or symbols. In this case, simply press the <OPTION> button and the <E> key simultaneously (<OPTION><E>) to enter the character editor. A green rectangular background will appear on the screen and an enlarged block layout of the current character (i.e. the one currently being displayed at the right in the text window) will be shown. A dark cursor will also be visible. You can move this cursor around the 8x8 grid of the enlarged character with the arrow keys. Pressing the space bar will "turn-on" a position that is "off", or "turn-off" a position that is presently "on". Each change is shown at the right of the text window. The entire character can be "cleared" by pressing <CTRL><CLEAR>. When you get the special character you want, you can end the editing session by simply pressing the <E> key. The special character just created will be available with the other special characters until the computer is turned off. You can also permanently save a special character by pressing the <CTRL><S> key while in the editing mode. This will replace the special characters which come with the program (which are stored on disk) by the ones you have created. (You may want to have different special symbol sets stored on different disks.)

A WORD OF WARNING - Some letters or characters may look very strange when entered as "small" characters. This is because the 7+ graphics display mode does not have enough resolution to adequately handle the normal ATARI character set printed in the "small" size. If you wish to enter small letters, press the <ESC> key to change the display mode to Basic graphics 8. This mode does have the resolution to display the small letters properly. (See Appendix A for a further discussion of display modes.)

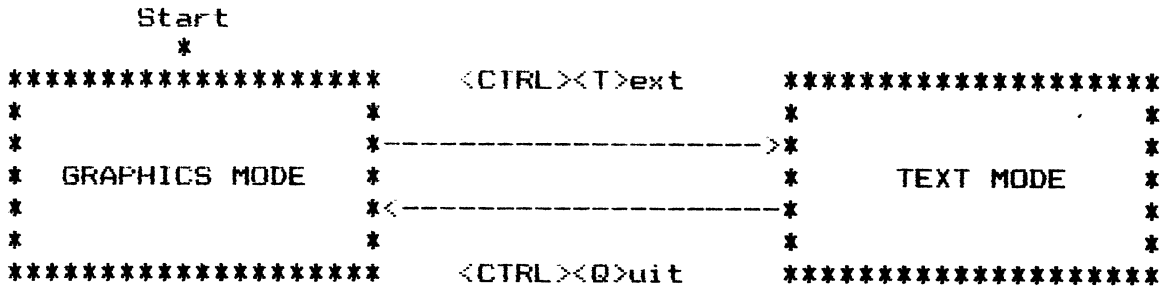
Moving Text Around the Screen

Finally, you may have a display you like, but some lettering is not centered or some piece is not quite in the desired location. In this case, you can use <CTRL><INSERT>, <CTRL><DELETE>, <SHIFT><INSERT>, and <SHIFT><DELETE> to move things around on the screen. These commands work in a manner similar to their operation in a normal text mode, BUT THEY NOW OPERATE ON THE ENTIRE GRAPHICS DISPLAY. For example, <SHIFT><INSERT> inserts one extra scan line at the location of the top of the cursor. Try out these commands and you will quickly get an idea of how they operate.

TROUBLE ???

If you ever get in a situation where you do not know what to do, press <SYSTEM RESET>. This should just restart the program WITHOUT erasing the screen display.

PROGRAM ORGANIZATION AND COMMAND SUMMARY



<L>ines (rubber-band lines)

<E>llipse
(Press <OPTION> to adjust shape of ellipse; press space bar when done)

<F>ill

<R>ectangle (mark opposite corners)

<C>olors (change colors)

<0>,<1>,<2>,or<3>
(color numbers)

<ESC> (change display graphics)

<P>rint display
(requires dot graphics printer -see Appendix C)

<S>ave display on disk

<G>et previously saved display from disk

<CTRL><CLEAR> or <SHIFT><CLEAR
(clears screen)

<D>raw (when fire button pressed)

<Z>oom (magnify display; fill pixel when fire button pressed; any other command to exit)

<X> (eXhibit screens; ie "slide show"; press <RETURN> when done

<CTRL><T> (go to text mode)

<BACK S>, <RETURN>, <CLEAR>, arrow keys, <CTRL><INSERT>, and <CTRL><DELETE> all work as in a normal typing mode.

<SHIFT><INSERT> inserts one scan line

<SHIFT><DELETE> deletes one scan line

<CTRL><U> moves cursor up 8 scan lines

<CTRL><D> moves cursor down 8 scan lines

<CTRL><F> goes forward through special character set

<CTRL><B> goes back through special character set

<TAB> first, then <1>,<2>,<3> for colors; <S>,<M>,<L> for text size

<OPTION><P> to print print special character

<OPTION><E> to edit special character (use arrow keys and space bar) <CTRL><S> saves new character to disk; <E> ends editing

<CTRL><Q> - return to graphics mode

Appendix A

ATARI HIGH RESOLUTION GRAPHICS DISPLAYS

One of the most impressive capabilities of the Atari is the wide variety of graphics display modes. This versatility is made possible by the use of a custom-designed microprocessor, called the ANTIC chip. ANTIC handles the screen display, freeing the 6502 microprocessor (and other chips) for other chores. In fact, the ANTIC chip even has its own program which tells it what kind of a screen display to generate. This program is called the display list. The ANTIC chip can actually display 6 types of character modes and 8 graphics modes (not counting the GTIA modes) and these modes can be mixed in a variety of ways. Atari BASIC, however, gives the programmer direct access to only 9 of these modes (called GRAPHICS 0-8 in BASIC).

Graphics 8 is the highest resolution mode available. In this mode there are 320 picture elements or "pixels" in each horizontal line across the screen. There are 160 such scan lines when the text window is present at the bottom of the screen. Each pixel can either be "on" or "off". One bit is thus required to tell the ANTIC chip whether or not to turn on the appropriate pixel. With 320 pixels per scan line and 8 bits per byte, it is easily seen that 40 bytes are required to describe each scan line. Each bit of any display memory byte is "mapped" directly into the pixels as shown in Figure A-1.

In BASIC, the next highest resolution display mode is Graphics 7. Graphics 7 has 160 pixels per scan line and 80 scan lines from the top of the screen to the text window. The ANTIC chip will also support another display mode, often called graphics 7+. This mode is identical with Basic's Graphics 7, except there are twice as many scan lines (i.e. 160 instead of 80). The vertical resolution is thus identical with Graphics 8. There are only half as many pixels in the horizontal direction, however, but each of these pixels can be given one of four colors (as compared to 1 color in Graphics 8). To keep track of the assigned color for each pixel, two bits are required. The screen data for four pixels is thus stored in one byte (see Figure A-1) , and 40 bytes are therefore needed for one scan line of display data. Only half as many pixels are diplayed in Graphics 7+ as in Graphics 8, but twice as many bits per pixel are required. This means that Graphics 7+ has the same memory requirements as Graphics 8. The only difference is how the ANTIC chip displays the data.

The same data can thus be displayed in Graphics 7+ or 8. One might expect the resulting displays to be quite different, however. This is true in many cases, but for some displays only the colors change when one flips from one display mode to another. This is due to an "accident" in color television performance called artifacting.

Artifacting is due the inablity of a standard color television to accurately display the 320 pixels per line of Graphics 8. The standard TV can display that many pixels but the apparent color of each pixel depends upon the pixel's location and whether or not nearby pixels are "on" or "off". Thus a random line drawn in Graphics 8 may seem to consist of several colors. As indicated in Figure A-2, the apparent color of a section of the screen will depend upon the pixel's location and whether the pixel next to it is also turned on. The actual colors seen depend upon the particular TV set. Note that the on-off bit patterns are similar to those that would appear in Graphics 7+ data (i.e. a (1 1) patern indicates color # 3 in 7+ and color A in Graphics 8). A little thought shows that multi-color images drawn in Graphics 7+ will also appear in multicolors (although different colors) in Graphics 8 due to artifacting.

You can investigate this effect by drawing a multi-colored picture in graphics 7+ and the changing the display to graphics 8 by pressing <ESC>. Several colors will be seen. If you then press <Z>oom to magnify the graphics 8 screen, only one color will be seen, but the alternating bit patterns shown in Figure A-2 will be apparent. Note that colors #1 and #2 will be printed in this alternating bit pattern form during a screen dump to the printer.

Artifacting can cause some problems, however. As previously mentioned, a slanted line may appear to be made up of several colors. Also, artifacting can make certain characters very hard to read. (Note that the Atari character set was carefully designed to minimize this problem by almost always having at least 2 horizontally adjacent pixels "on" in every scan line of every character.)

You can see these effects easily with the ALOG DISPLAYMAKER. First draw a few lines or circles in different colors in the 7+ mode. (The program starts out in this mode.) Then flip to Graphics 8 by pressing the <ESC> key. Now draw some more lines or circles. The lines drawn in Graphics 8 have a finer resolution but are multicolored. You may thus want to use Graphics 7+ for displays where color uniformity is important.

Unfortunately, Graphics 7+ does not have the resolution to display small text characters, such as those used in Basic's Graphics 0. These characters are 8 pixels wide and 40 characters can be put on one line. It thus requires a resolution of 320 pixels per scan line to display these characters. This is achieved only in Graphics 8. Each byte of a character is mapped into only 4 pixels in Graphics 7+. The result is that small letters may look very stange in Graphics 7+. Note that the three colors of small text letters that can be typed with the ALOG DISPLAYMAKER are achieved by the use of artifacting. Some of these characters are distorted; this can be eliminated by changing to color <3> before typing the character. Also, there is an upper case alphabet in the special character set that will look OK in Graphics 7+. Each character is only 4 Graphics 7+ pixels wide, however, so that some characters are hard to distinguish. All of these effects make Graphics 7+ more useful for some things and Graphics 8 better for others. A good strategy in using the ALOG DISPLAYMAKER may be to use 7+ for making geometric drawings (e.g. circles, lines, etc.) and then change to Graphics 8 for adding small text.

One additional difference between modes 7+ and 8 should be noted. Graphics mode 7+ is a true 4 color mode with the colors of each pixel determined by 4 color registers (labeled 0-3 here with 0 being the background color). You must therefore change all four registers (with the <C>olors command) to completely change the colors seen on the screen. In Graphics 8, on the other hand, you can "see" four different colors, but they are not really independent. In fact, the colors seen on the screen are totally determined by only two color registers (#2 and #3, as labeled here). If you try changing colors while in the Graphics 8 mode you can rapidly get an idea of how these two registers control the color and luminance of the display.

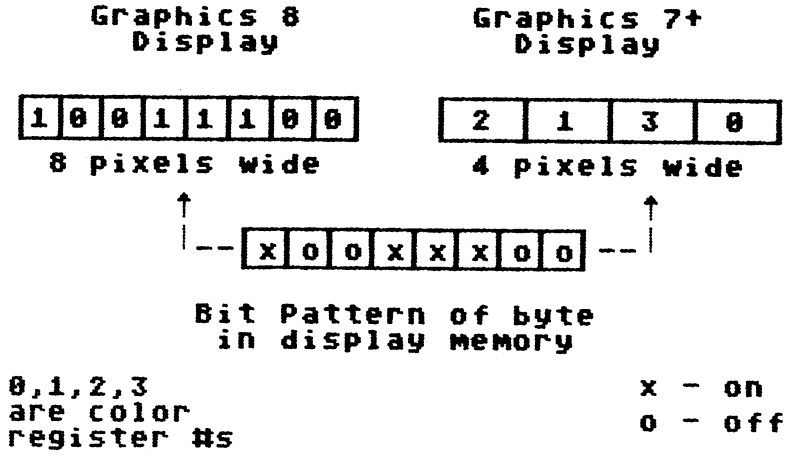


FIGURE A-1

ARTIFACTING IN GRAPHICS 8

11111111 - Solid line (Color "A")

10101010 - Will "appear" as solid line with artifact (Color "B")

01010101 - Will "appear" as solid line with artifact (Color "C")

Bit or pixel patterns

FIGURE A-2

APPENDIX B

USING DISPLAYMAKER SCREENS AND CHARACTER SETS IN BASIC PROGRAMS

The ALOG DISPLAYMAKER is written in a computer language called valFORTH, sold by Valpar International, 3801 E. 34th Street, Tucson, AZ 85713 (602 790-7141). This is a version of the very powerful FORTH language, enhanced to use the many special features of the Atari computer. FORTH is a very versatile language that runs much faster than BASIC, yet is quicker to program in than assembly language. One of the particular features used here is the direct reading and writing of disk sectors. Atari DOS is thus not used to store or retrieve any display screens. One can thus store more data on a disk, but such data cannot be read using normal DOS techniques.

However, it is relatively easy to read data from a disk without using DOS. The attached BASIC program contains an assembly language subroutine (in the data statements) which reads one sector from a disk in drive #1. Remember that there are 720 sectors and each sector contains 128 bytes of data.

Each screen stored by the ALOG DISPLAYMAKER occupies 50 sectors of the disk. Screen #1 starts at sector #301, screen #2 at #351, etc. The BASIC program given here shows how to load this data directly into a Graphics 8 display.

Note that this method of storing data does not use the "Volume Table of Contents" normally stored on sector #360 by Atari DOS. In fact, screen #2 uses this sector for storage. Because Atari DOS looks at this sector when copying a disk, you may be unable to copy an ALOG DISPLAYMAKER disk after you have stored a display at location #2. Your original program disk was initialized to avoid this problem. You should avoid storing data on screen #2 of this disk so that you will not have problems in making copies. You can also re-initialize this sector to zeroes by simply storing a blank display in screen #2.

You may also want to use a special character set (created with the character editor) in some other program. The 64 special characters are stored in sectors 71, 72, 73, and 74. Each character requires 8 bytes, so that 16 characters are stored in each sector. Character #1 is the one seen when the text mode is accessed for the first time. It is described by the first 8 bytes in sector 71. Other characters are sequentially stored.

```
1 REM ************************************************************
2 REM ****   EXAMPLE OF HOW TO LOAD A SCREEN CREATED WITH THE ALOG    ****
3 REM ****              DISPLAYMAKER USING A BASIC PROGRAM             ****
4 REM ****                                                             ****
5 REM ****   AN ASSEMBLY LANGUAGE SUBROUTINE ( CONTAINED IN DATA       ****
6 REM ****   STATEMENTS AND PUT IN A STRING ) IS GIVEN FOR READING     ****
7 REM ****   ONE SECTOR OF THE DISK AT A TIME                          ****
8 REM ****                                                             ****
9 REM ****             USE IN THE FORM:                                ****
10 REM ***                                                             ****
11 REM ***  X=USR(subroutine address,starting address of destination, ****
12 REM ***        sector number)                                       ****
13 REM ***                                                             ****
14 REM ************************************************************
200 REM *** PUT ROUTINE TO READ A SECTOR IN THE STRING SECRD$ ***
210 DIM SECRD$(36)
220 FOR I=1 TO 36
230 READ X
240 SECRD$(I,I)=CHR$(X)
250 NEXT I
300 REM *** USE SCREEN # 1 AS EXAMPLE ***
310 GRAPHICS 8:SCRNUM=1
320 START=PEEK(88)+256*PEEK(89)
321 REM * START IS THE ADDRESS CORRESPONDING TO TOP LEFT CORNER OF THE
322 REM GRAPHICS 8 SCREEN *
330 FOR I=0 TO 49
340 DESTADDR=START+I*128
341 REM * DESTADDR IS THE ADDRESS WHERE WE WANT THE 128 BYTES FROM
342 REM SECTOR BEING READ TO START *
350 SECNUM=301+(SCRNUM-1)*50+I
351 REM * SECNUM IS THE NUMBER OF THE DISK SECTOR BEING READ *
360 X=USR(ADR(SECRD$),DESTADDR,SECNUM)
370 NEXT I
400 END
1000 DATA 104,104,141,5,3,104,141,4,3,104
1010 DATA 141,11,3,104,141,10,3,169,1,141
1020 DATA 1,3,169,82,141,2,3,32,83,228
1030 DATA 173,3,3,133,212,96
```

APPENDIX C

SENDING DISPLAYS TO A PRINTER

It is often very useful to be able to make a printed copy of a screen display. The ALOG DISPLAYMAKER has the ability to print copies of the screen in two different sizes, using dot matrix printers with bit-plot graphics capabilities. Examples of such printers include the EPSON (with graftrax) and the PROWRITER . Such printers operate by printing a dot if a certain bit is "on" or not printing the dot if the bit is "off". In general, 8 bits (1 byte) are sent to the printer at a time, so that 8 dots can be controlled. These dots usually (if not always) correspond to 8 vertical print heads. A one dot wide column up to 8 dots high can thus be printed with one byte of data. Unfortunately, there are several different methods used to determine which print head is controlled by which bit and different printers use different control codes to tell them to enter a bit-plot graphics mode.

Your original ALOG DISPLAYMAKER disk will print properly with an EPSON MX-80 printer with the Graftrax upgrade or a Gemini 10 printer. If you hit the <P> key while in the graphics mode, you will be asked if you want a quick (small) print-out or a double size print-out. The small size print-out is 160 dots across the page and 320 dots vertically. It thus corresponds directly to the display screen turned 90 degrees (so that the long axis of the screen corresponds to the long axis of a normal sheet of paper). The small screen dump plot will occupy about one-quarter of an 8 1/2 x 11 page. The double size print-out is simply twice as big. (320 x 640). It will approximately fill one standard page on an EPSON MX-80.

The ALOG DISPLAYMAKER is designed so that the user can alter the printer control codes so that displays can be dumped on other printers. To alter the program, first enter the print section by pressing <P> while in the graphics mode. To change the control codes, then press <CTRL><C> instead of <RETURN> or <D>. You will then be asked to enter various information needed by the program.

You will first be asked whether the top print head (top dot) corresponds to the most significant bit (MSB) of the data byte. If your printer follows this convention, then a byte equal to 128 would be printed as one dot with seven blanks below it. In some printers (such as the PROWRITER) the top dot is the least significant bit. A value of 128 would thus be printed as 7 blanks arranged vertically above a single dot. If you do not answer "Yes" (<Y>) to this question, the program assumes that the least significant bit corresponds to the top print head. Note also that some printers can be set to use only 7 of the 8 bits in a data byte. This program use all 8 bits and internal switches should be set appropriately.

You will then be asked to "TYPE PRE-DUMP COMMANDS". Such commands include any command needed to prepare your printer for bit-plot graphics. One command almost certainly needed is the one to set the line feed spacing. You may also want to set the horizontal spacing between dots (i.e. dots per inch). All commands are entered in string form, with each character printed representing one byte sent to the printer. Pre-dump commands are sent before the actual screen dump process begins. You can enter up to 36 commands. This command string should be terminated with at least one blank. Then hit <RETURN> to send the string to memory.

For example, if you have a PROWRITER, you can first set the pitch to Pica (10 CPI) and then set the line spacing to 16/144 inches by hitting the keys (Each [ ] indicates a single character)

[<ESC>] [<N>] [<ESC>] [<T>] [<1>] [<6>] [<space bar>]

You will then be asked to "TYPE COMMANDS FOR 160 x 8 PRINT". These are the commands necessary to print one line of the small screen dump (i.e. 160 columns, each 8 dots high). As previously, commands are entered in string form and must end with a blank. Again using the PROWRITER as an example, one would enter

[<ESC>] [<S>] [<0>] [<1>] [<6>] [<0>] [<space bar>]

Finally, you must enter the commands to print 320 bit-mapped graphics characters on one line when asked "TYPE COMMANDS FOR 320 x 8 PRINT". For the PROWRITER, this is

[<ESC>] [<S>] [<0>] [<3>] [<2>] [<0>] [<space bar>]

Once these commands have been entered, they are stored on the program disk and will be loaded with the program. You thus need to customize the program to your specific printer only once.

As previously mentioned, the ALOG DISPLAYMAKER program disk is initialized to use an Epson printer with the graftrax option. If you should ever change the printer control codes and want to go back to the Epson, the appropriate responses are:

```
IS TOP DOT MSB ?      [ <Y> ]
    (Y/N)

TYPE PRE-DUMP COMMANDS     [<ESC>][<A>][<CTRL><H>][<space bar>]

TYPE COMMANDS FOR 160 x 8 PRINT     [<ESC>][<K>]
                                    [<Atari key><space bar><Atari key>]
                                    [<CTRL><,>][<space bar>]
```

(Note that [<Atari key><space bar><Atari key>] simply shifts the keyboard to inverse video, prints an inverse space,and then shifts back to the normal (non-inverse) display. Also [<CTRL><,>] just prints a heart.)

TYPE COMMANDS FOR 320 X 160 PRINT	[<ESC>][<K>][<SHIFT><8>] [<CTRL><A>][<space bar>]